



# Spatial Reduction Algorithm for Atmospheric Chemical Transport Models

## Citation

Rastigeyev, Y., Michael P. Brenner, and Daniel J. Jacob. 2007. Spatial reduction algorithm for atmospheric chemical transport models. *Proceedings of the National Academy of Sciences* 104(35): 13875-13880.

## Published Version

doi:10.1073/pnas.0705649104

## Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:4724833>

## Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA>

## Share Your Story

The Harvard community has made this article openly available.  
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

# Spatial reduction algorithm for atmospheric chemical transport models

Y. Rastigejev, M. P. Brenner\*, and D. J. Jacob

School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02139

Communicated by James R. Rice, Harvard University, Cambridge, MA, June 19, 2007 (received for review January 17, 2007)

**Numerical modeling of global atmospheric chemical dynamics presents an enormous challenge, associated with simulating hundreds of chemical species with time scales varying from milliseconds to years. Here we present an algorithm that provides a significant reduction in computational cost. Because most of the fast reactants and their quickly decomposing reaction products are localized near emission sources, we use a series of reduced chemical models of decreasing complexity with increasing distance from the source. The algorithm diagnoses the chemical dynamics on-the-run, locally and separately for every species according to its characteristic reaction time. Unlike conventional time-scale separation methods, the spatial reduction algorithm speeds up not only the chemical solver but also advection–diffusion integration. Through several examples we demonstrate that the algorithm can reduce computational cost by at least an order of magnitude for typical atmospheric chemical kinetic mechanisms.**

atmospheric chemistry | chemical kinetics | multiscale analysis

The modeling of tropospheric oxidants (ozone and the hydroxyl radical OH) is of central importance for addressing issues of air quality, aerosol formation and evolution, acid formation, and global budgets of greenhouse gases such as methane (1). Although the chemical factors controlling tropospheric ozone and OH are fairly well established, the computational challenge of modeling concentrations in relation to changes in emissions or climate is enormous. Current photochemical mechanisms describing oxidant chemistry include hundreds of coupled chemical species reacting on time scales ranging from milliseconds to many years (2).

Typical chemical transport models (CTM) combine chemical reactions with advection by a meteorologically predicted flow velocity. The resulting system of equations is extremely stiff, nonlinear and involves a large number of chemically interacting species. The difficulty of solving these equations imposes severe limitations in the spatial resolution of the CTMs, as well as the timescales that can be simulated. This difficulty represents a major obstacle for progress in atmospheric chemistry research. Even with a simplified chemical mechanism of  $\approx 100$  species, present computational resources limit the number of CTM grid points to  $\approx 10^5$  to  $10^6$ , corresponding to a horizontal resolution  $\approx 100$  km and a vertical resolution  $\approx 1$  km (3). This is not adequate for resolving the vertical stratification of the troposphere (4), the dynamics of the planetary boundary layer (5), convective processes (6), or sharp horizontal gradients across fronts and coastlines (7). The push for fast, high-resolution CTMs will be exacerbated over the next decade by the need to interpret satellite observations of tropospheric ozone and related species (8).

In this paper, we present a numerical algorithm that addresses two major aspects of this problem: the large number of interacting chemical species and extremely large stiffness of the chemical model. Our algorithm is inspired by the observation that atmospheric emissions have most of their fast reactants and their quickly decomposing reaction products localized near the emitter, typically near the ground. Far from the emitter, the fast reactants do not play a significant role. There is thus the opportunity to spatially reduce the chemical mechanism, where a full chemical mechanism is used

near emission sources and reduced mechanisms are used far from sources.

To give an idea of the scale of the computational savings that could result from such an algorithm, consider the chemical mechanism of the GEOS-Chem CTM (9), extensively used for tropospheric chemistry applications (10):  $\approx 65$  of the 120 species describe nonmethane hydrocarbons (NMHC) and their oxidation products. These NMHCs are emitted by anthropogenic and natural processes at the surface of continents. Most have short lifetimes. The most important NMHC is biogenic isoprene, which has a lifetime of  $< 1$  h against oxidation and whose oxidation products account for 25 species in the mechanism. Although proper simulation of isoprene and other NMHC chemistry is critical near the ground (11), in the remote troposphere, concentrations of NMHCs and their reactive oxidation products are in general very low and a simpler chemical mechanism including 15 species can be sufficient (12).

The idea of reducing chemical mechanisms is quite old. Traditional methods address spatially uniform situations, and exploit the large range of time scales between different chemical reactions, by splitting the species ensemble into subsets that evolve over similar time scales (13–19). This splitting speeds up computation by reducing the size and the stiffness of the individual Jacobian matrices to be inverted.

The spatial reduction mechanism proposed here gradually and continuously discards the equations for fast reacting species with distance from a pollution source. We track a series of chemical boundary layers (CBL), which separate regions where the concentration of a given species is computed with its equation of motion from those where the concentration is determined by extrapolation from the source. The number of domains (multiconnected in general) can be as large as the number of chemical species. The variable CBLs represent the region of the atmosphere affected by fresh emissions; they include not only the meteorological boundary layer but also possibly pollution plumes injected to high altitudes.

## The Chemical Mechanism

Emissions of NMHCs and  $\text{NO}_x$  are the principal drivers of complexity in tropospheric oxidant chemistry. The former are emitted by vegetation, combustion and industrial processes whereas the latter are emitted by combustion, soils and lighting. Because the oxidation schemes of all NMHCs follow similar schematics (20), we use propene ( $\text{C}_3\text{H}_6$ ) as representative NMHC for a simplified representation of the chemistry, and include only the most important reactions in its oxidation sequence. This greatly decreases the number of species in the chemical mechanism while retaining the essence of the mechanism complexity.

Table 1 lists the reactions in the reduced mechanism, and Fig. 1 illustrates the major features of the reaction scheme. Oxidation of

Author contributions: Y.R., M.P.B., and D.J.J. designed research; Y.R., M.P.B., and D.J.J. performed research; Y.R., M.P.B., and D.J.J. analyzed data; and M.P.B. and D.J.J. wrote the paper.

The authors declare no conflict of interest.

Abbreviation: CTM, chemical transport model.

\*To whom correspondence should be addressed. E-mail: brenner@seas.harvard.edu.

© 2007 by The National Academy of Sciences of the USA

**Table 1. Reduced propene –  $NO_x$  –  $HO_x$  –  $O_3$  atmospheric chemistry mechanism**

Reaction*	Reaction rate <sup>†</sup>
$O_3 + h\nu \xrightarrow{H_2O} 2OH$	$10^{-6}$
$O_3 + NO \rightarrow NO_2 + O_2$	$2. \times 10^{-14}$
$O_3 + OH \rightarrow HO_2 + O_2$	$7.3 \times 10^{-14}$
$O_3 + HO_2 \rightarrow HCHO + HO_2 + 2O_2$	$10^{-17}$
$HO_2 + NO \rightarrow OH + NO_2$	$8.1 \times 10^{-12}$
$HO_2 + HO_2 \rightarrow H_2O_2 + O_2$	$2.1 \times 10^{-12}$
$NO_2 + h\nu \xrightarrow{O_2} NO + O_3$	$8.8 \times 10^{-3}$
$NO_2 + OH \xrightarrow{M} HNO_3$	$2.5 \times 10^{-11}$
$HCOH + OH \xrightarrow{O_2} CO + HO_2 + H_2O$	$9. \times 10^{-12}$
$CO + OH \xrightarrow{O_2} HO_2 + CO_2$	$2.5 \times 10^{-13}$
$C_3H_6 + OH \xrightarrow{O_2} PO_2$	$3. \times 10^{-11}$
$PO_2 + NO \xrightarrow{O_2} NO_2 + HO_2 + HCHO + ALD_2$	$8.7 \times 10^{-12}$
$ALD_2 + OH \xrightarrow{O_2} MCO_3 + H_2O$	$1.4 \times 10^{-11}$
$MCO_3 + NO_2 \xrightarrow{M} PAN$	$1.1 \times 10^{-11}$
$PAN \xrightarrow{M} MCO_3 + NO_2$	$1.3 \times 10^{-3}$
$HO_2 + MCO_3 \rightarrow MAP + O_2$	$1.3 \times 10^{-11}$
$OH + MAP \rightarrow 0.5OH + 0.5HCHO + 0.5MCO_3 + H_2O$ $+ 0.5CO_2$	$7.4 \times 10^{-12}$
$MO_2 + NO \xrightarrow{O_2} HCHO + HO_2 + NO_2$	$7.7 \times 10^{-12}$
$MO_2 + HO_2 \rightarrow MP + O_2$	$4.6 \times 10^{-12}$
$MP + OH \rightarrow MO_2 + H_2O$	$5.2 \times 10^{-12}$
$MCO_3 + NO \rightarrow MO_2 + NO_2 + Co_2$	$2. \times 10^{-11}$
$HCHO + h\nu \xrightarrow{2O_2} 2HO_2 + CO$	$1. \times 10^{-4}$
$MP + h\nu \xrightarrow{O_2} HCHO + HO_2 + OH$	$1. \times 10^{-5}$
$OH + HO_2 \rightarrow H_2O + O_2$	$1.1 \times 10^{-10}$

\*Reactions and rate constants are extracted from the GEOS-Chem mechanism. Original references are given in the GEOS-Chem chemical mechanism documentation (28). Rate constant values (in units of  $\text{sec}^{-1}$  or  $\text{m}^3/\text{molecules}/\text{sec}$ ) are computed for a temperature of  $T = 298 \text{ K}$  and a pressure of 1 atm. The following shorthand notation is used:  $\text{PO2} = \text{CH}_3\text{G}(\text{OH})\text{CH}_2\text{O}_2$ ,  $\text{ALD2} = \text{CH}_3\text{CHO}$ ,  $\text{MCO3} = \text{CH}_3\text{C}(\text{O})\text{OO}$ ,  $\text{MO2} = \text{CH}_3\text{O}_2$ ,  $\text{PAN} = \text{CH}_3\text{CO}(\text{OONO}_2)$ ,  $\text{MAP} = \text{CH}_3\text{G}(\text{O})\text{OOH}$ ,  $\text{MP} = \text{CH}_3\text{OOH}$ .

<sup>†</sup>Reactions with species above the reaction bar are multistep processes in which the species above the bar are not involved in rate limiting steps and thus do not contribute to the rate calculation. Letter "M" above the reaction bar indicates any inert molecule (in the atmosphere, mainly  $N_2$  and  $O_2$ ) that stabilizes a reaction product (three-body reaction) or drives thermolysis.

propene by OH produces a suite of carbon compounds leading eventually to CO<sub>2</sub>. The time scales of the reactions vary from seconds (lifetime of OH) to minutes (lifetimes of PO<sub>2</sub>, MCO<sub>3</sub>, MO<sub>2</sub>) to hours (lifetimes of other propene intermediates) to weeks or greater (CO).

The evolution of  $n_i(x, t)$ , the number density of the  $i$ th chemical species, couples chemical reactions with advection and diffusion through the continuity equation

$$\frac{\partial n_i}{\partial t} + \nabla \cdot (\mathbf{u} n_i - \kappa \nabla n_i) = \dot{\omega}_i + s_i(\mathbf{x}, t) \quad i = 1 \cdots N, \quad [1]$$

where  $\mathbf{u}$  is the wind velocity,  $\kappa$  is the turbulent diffusivity,  $\dot{\omega}_i$  is the production rate of the  $i^{\text{th}}$  chemical species, and  $s_i$  describes local emissions and nonchemical sinks. The local net production rate  $R(i) \equiv \dot{\omega}_i + s_i(\mathbf{x}, t)$  of  $i^{\text{th}}$  species can be separated into the local gross production rate  $R^+(i)$  and the local loss  $R^-(i)$  terms, i.e.,  $R(i) = R^+(i) - R^-(i)$ . Emissions are in general near the ground but can extend to all altitudes (aircraft, lightning, volcanic emissions, etc.).

## Spatial Reduction Algorithm

**Introduction.** The spatial reduction algorithm partitions the computational domain for every chemical species into a “fast region” where we calculate the species concentration by solving the full system of equations, and a “slow region” where we calculate the species concentration by extrapolating from the “fast region.” Because the analysis is done on a species-by-species basis, the spatial reduction is gradual with fast equations being eliminated closer to the source.

We explain our algorithm in the context of a very simple example. Consider a three species reaction system, comprised of chemicals  $A$ ,  $B$ , and  $C$ . The reactant  $A$  decays quickly through  $A + B \rightarrow C$ . The concentration of  $B$  is assumed fixed, and there is a (slowly changing) flux  $f(t)$  of  $A$  at the ground. We assume that  $A$  is the fastest reactant, so that  $\tau_C, \tau_B \gg \tau_A$ , where  $\tau_B$ ,  $\tau_C$ , and  $\tau_A$  are the characteristic decay times for the elements  $B$ ,  $C$ , and  $A$  respectively. Additionally, we assume that the only transport mechanism is 1- $D$  diffusion.

Under these conditions, the concentration of  $A$  is localized near the ground. The dynamics of  $A$  and  $C$  are described by

$$\frac{\partial n_A}{\partial t} = -\frac{n_A}{\tau_A} + \kappa \frac{\partial^2 n_A}{\partial x^2} \quad [2]$$

$$\frac{\partial n_C}{\partial t} = -\frac{n_C}{\tau_C} + kn_B n_A + \kappa \frac{\partial^2 n_C}{\partial x^2}, \quad [3]$$

where  $\kappa$  is the diffusion coefficient, and the boundary conditions are  $-\kappa \partial_x n_A|_{x=0} = f(t)$ . As long as  $f(t)$  varies more slowly in time than  $\tau_A$ , the solution to Eq. 2 is approximately

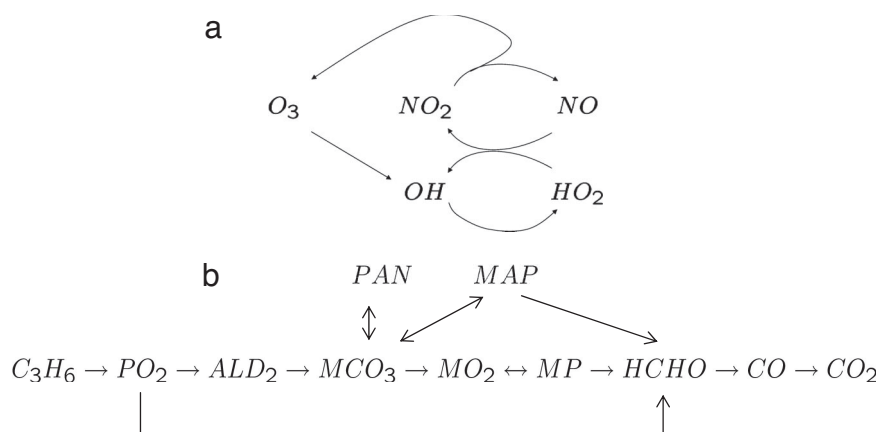
$$n_A(x, t) = f(t) \sqrt{\frac{\tau_A}{\kappa}} \exp(-x/\sqrt{\kappa\tau_A}). \quad [4]$$

This equation for  $n_A(x, t)$  can then be used in Eq. 3, so that we now have a single equation for  $n_C(x, t)$  which is valid far from the source. This represents a (simple) spatial reduction because we now need to solve only one equation far from the ground. This equation is

$$\frac{\partial n_C}{\partial t} = -\frac{n_C}{\tau_C} + kn_B\gamma \exp(-\alpha x) + \kappa \frac{\partial^2 n_C}{\partial x^2}. \quad [5]$$

**Eq. 5** uses  $\gamma$  and  $\alpha$  as free constants, not necessarily given by  $\gamma = f(t)\sqrt{\tau_A/\kappa}$  and  $\alpha = 1/\sqrt{\kappa\tau_A}$  implied by **Eq. 4**. This is to emphasize that in more general situations, it is not possible to analytically solve for the fast reactant profiles. However,  $\gamma$  and  $\alpha$  can always be determined by using matching conditions, by requiring that both concentrations and fluxes are continuous at the boundary between the fast and slow region. Thus, if  $n_{A,f}$  is the concentration of species  $A$  as computed by the full solver in the fast region, and the boundary between fast and slow occurs at  $x = x_b$ , we require that  $\gamma = n_{A,f}(x_b)$ , and  $\alpha = -\partial_x n_{A,f}(x_b)/n_{A,f}(x_b)$ . This method for choosing  $\gamma$  and  $\alpha$  ensures that the spatially reduction conserves mass exactly.

This one-dimensional example illustrates the basic principles of spatial reduction: in the fast region, we solve the full chemical mechanism, whereas in the slow region we extrapolate concentrations of the fast reactants from the fast region. In the present example the full reaction system is comprised of two coupled partial differential equations (PDE), whereas the reduced system involves solving only one PDE. The computational cost of solving a system



**Fig. 1.** Schematic of the reduced chemical mechanism in Table 1. (a) Cycling of  $HO_x$  radicals,  $NO_x$  radicals, and ozone. The organic peroxy radicals in the mechanism ( $PO_2$ ,  $MCO_3$ ,  $MO_2$ ) perform functions similar to  $HO_2$ . (b) Intermediate species in the oxidation of propene to  $CO_2$ . The concentrations of all 15 species shown in this figure (excepting  $CO_2$ ) are computed within the chemical mechanism.

of PDEs using an implicit solver increases at least quadratically with the number of dependent variables. In the present simple example, the CPU time reduction will thus be a factor of four. We note that, in practice, the CPU savings should be greater because by eliminating the fast variables we are eliminating the primary source of stiffness.

**Domain Partitioning.** How do we choose the borderline between the fast and slow region? Let us denote the fast and slow regions as  $D_f$  and  $D_s$ , respectively, with the regions separated by a moving boundary  $\Gamma$ . In the above example,  $\Gamma$  corresponds to the point  $x_b$ . It is intuitively clear that in the fast region  $D_f$  the species density  $n$  should either be sufficiently large, or have the potential to become large in a short period since its production rate is large. Our domain partitioning criterion thus relies on the local magnitudes of  $n$  and the local net  $R(i)$  or gross production  $R^+(i)$  rates. Because  $R(i) \leq R^+(i)$ , the criteria based on the local gross production rate  $R^+(i)$  is more conservative than the one based on the local net production  $R(i)$ . One definition of  $D_f$  uses an absolute threshold for  $n(i)$  and  $R(i)$ : for each chemical species  $n(i)$  we require that the chemical concentration  $n(i) > n(i)_0$  and the reaction rate  $R(i) > R(i)_0$  or  $R^+(i) > R(i)_0$ , where  $n(i)_0$  and  $R(i)_0$  are prescribed thresholds. Alternatively, we can define  $D_f$  using relative thresholds. Here, we first find the local maxima of  $n(i)$  and  $R(i)$  ( $R^+(i)$ ), and then define  $D_f$  by requiring that  $n(i)$  and  $R(i)$  are larger than some fraction  $\delta$  of their values at the local maximum. The computations presented below use a relative threshold.

**Matching Conditions and Extrapolation Algorithm.** We discussed above, how we match the solution between the fast domain and the slow domain by imposing continuity of chemical concentrations and fluxes. In the one-dimensional example these conditions fixed the parameters  $\alpha$  and  $\gamma$ , which allowed extrapolation of the fast species into the slow region through exponential decay. There are two additional complications in multidimensions.

First, our argument for the exponential decay of the fast reactants into the slow region was in the context of a one dimensional model with eddy viscosity. One might question whether this would still hold in multidimensions with advection. In fact, the exponential decay of the fast reactant is a generic feature of a chemical source in multidimensions, with transport by either diffusion or advection. To see this, consider an isolated chemical source of strength  $q$  located at the origin. The air is moving with a local velocity  $u$  along the  $x$  axis and the value of turbulent diffusion coefficient is  $\kappa$ . Assume that the chemical decays with characteristic reaction time  $\tau$ . The concentration  $n$  satisfies the continuity equation

$$u \frac{\partial n}{\partial x} = \kappa \nabla^2 n - \frac{n}{\tau} + q \delta(\mathbf{r}), \quad [6]$$

which has the exact solution

$$n = \frac{q}{4\pi\kappa r} \exp(-r(\sqrt{u^2 + 4\kappa/\tau} - u \cos \phi)). \quad [7]$$

Here,  $\phi$  is an angle between  $r$  and  $x$ . Because exponential decay dominates over geometric decay  $1/r$  far from the source, the concentration decays exponentially. This derivation assumes that the chemical source is steady; in practice this means that as long as the source varies on a time scale much longer than the chemical decay time  $\tau$ , this derivation holds.

The second major complication in multidimensions is that now the matching conditions must be satisfied on the surface separating the fast region and the slow region. Therefore, we need to find a consistent method for extrapolating information on this surface into the slow region through the expected exponential decay. Stated mathematically, given a chemical species denoted by  $g$  defined on  $\Gamma$ , we need to extrapolate  $g$  so that it decays exponentially with distance from  $\Gamma$ , with the decay rate

$$\alpha_n = -\frac{1}{g} \frac{\partial g}{\partial n} \Big|_r, \quad [8]$$

determined by the rate of change of the  $g$  normal to the interface. Note that  $\alpha_n$  is not constant, but varies over the interface  $\Gamma$ .

The problem of efficiently extrapolating information defined at an interface to a full domain has been previously addressed extensively in the context of the level set method (21). We first define a function  $d$ , which measures distance from  $\Gamma$  in the region we wish to extrapolate. As a distance metric, we require that  $|\nabla d| = 1$ ; additionally it must vanish on the interface ( $d|_\Gamma = 0$ ). Then we define the set of curves  $\Gamma^\perp$  that includes the boundary interface  $\Gamma$  as well as other equally distanced from  $\Gamma$  curve i.e.,  $\{\Gamma|d = \text{const}\}$  as well as the set  $\Gamma^\perp$ , which are locally perpendicular to  $\Gamma$  (see Fig. 2). To extrapolate  $\alpha_n$  from  $\Gamma$  to the entire region, we require that  $\alpha_n$  is constant along each curve of  $\Gamma^\perp$ , and hence that on these curves  $g$  decays exponentially. Because the normal vector  $n = \nabla d$  to  $\Gamma$  is tangent to  $\Gamma^\perp$ , these requirements are equivalent to the equations

$$\nabla d \cdot \nabla \alpha_n = 0 \quad [9]$$

$$\nabla d \cdot \nabla g + \alpha_n g = 0. \quad [10]$$



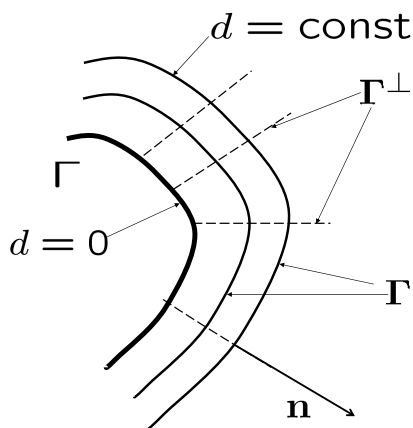


Fig. 2. Schematic picture of the level set curves. The boundary interface  $\Gamma$  is depicted by thick solid line and the two level set curves corresponding to different values of  $d$  are depicted by thin solid lines. The curves  $\Gamma^\perp$  are shown as dashed lines.

Eq. 9 states that the gradient of the local exponential decay rate is orthogonal to the local normal vector  $n$ , so that  $\alpha_n$  is constant along each curve of  $\Gamma^\perp$ . Eq. 10 requires that  $g$  exponentially decays with rate  $\alpha_n$  along  $\Gamma^\perp$ . Eqs. 9 and 10 can be solved simultaneously and explicitly by marching outwards from the separating interface  $\Gamma$  into the domain  $D_s$  using the Fast Marching Method (22).

**Computational Savings.** How much will the spatial reduction algorithm reduce computation time in the full problem? To estimate the savings, let us assume that there are  $N$  reacting species with  $N_s$  slow-reacting species, that the volume of the computational domain is  $V$  and that of the fast reacting domain is  $V_f$ . An implicit solver needs matrix inversion, which causes computational cost increase quadratically or even faster with the number of species. Assuming a constant mesh spacing, the reduced system has a fraction

$$\phi = V_f/V + \left(\frac{N_s}{N}\right)^2 (1 - V_f/V) \quad [11]$$

of the number of equations of the original full system. For the GEOS-Chem mechanism, we expect both  $V_f/V$  and  $N_s/N$  to be of order one tenth, hence implying an order of magnitude expected speed up in the computation time. In making this estimate we have used the fact that the spatial reduction does not substantially increase the CPU time; i.e., the computational cost of Eqs. 9 and 10 integration with described above explicit method is negligible (which is indeed true), and have also neglected additional benefits related to stiffness reduction.

**Summary of Algorithm.** To summarize, the numerical algorithm consists of three stages to implemented at each timestep. First, we construct a domain partitioning of every individual species based on the current species density  $n(\mathbf{x})$  and source  $S(\mathbf{x})$  distributions. This domain partitioning implies that there are different numbers of equations (and variables) to be integrated at each grid point. In the second stage, we integrate the fast equations at each grid point to advance the solution to the next time step. Finally in the third stage, we extrapolate the fast species concentration into the slow domains. Note that the domain partitioning is recomputed at every time step, so that the distribution of fast and slow species changes dynamically throughout the simulation.

In the test simulations that follow, we advance Eq. 6 using second-order Strang operator splitting for the time integration, separating the advection–diffusion part of the calculation from the reaction part. The advection diffusion calculation uses second order spatial central differences and the linearized trapezoidal method for

time stepping; the reaction part of the problem is solved by using a Bulirsch–Stoer (23) scheme.

## Numerical Experiments

We now proceed to demonstrate the implementation of the algorithm in both one and two spatial dimensions.

**One-Dimensional.** We first consider the performance of the algorithm in an example with one spatial dimension. To demonstrate the algorithm's ability to locate and adapt to pollution sources, we choose an example where the location of the pollution source changes in time. The challenge here is that the algorithm must automatically detect the regions where a new pollutant plume occurs.

We consider an isothermal atmosphere ( $T = 298$  K), and have the chemical transport take place in the altitude range  $0 < x < L$ , where  $L = 15$  km. The computational domain is uniformly resolved with 100 points. The initial concentration of all species is taken to be 0 except  $n_{O_3} = 10^{12}$  molecules per  $\text{cm}^3$ . We impose zero flux at the edges of the domain for all species except the pollutants  $\text{C}_3\text{H}_6$  and  $\text{NO}$ , where the fluxes at the ground are  $J_{\text{C}_3\text{H}_6} = 10^{11}$  molecules per  $\text{s cm}^2$  and  $J_{\text{NO}} = 10^{11}$  molecules per  $\text{s cm}^2$ . We take the turbulent diffusion coefficient  $\kappa = 3 \times 10^5 \text{ cm}^2/\text{s}$ .

In addition to the stationary source, we also implement a simple model of a thunderstorm partway through the simulation. Our thunderstorm rapidly transports pollutants from the lower to the upper troposphere. We assume that the thunderstorm starts at the time  $t_0 = 3 \times 10^4$  s and ends at time  $t_1 = 5 \times 10^4$  s. The thunderstorm is modeled by a pollutant sink that scavenges chemicals in the lower troposphere (defined to be  $0 \leq x \leq x_0 = 1.5$  km) and instantly transports them in a “chimney” upwards by a distance  $x_1 = 9$  km. The pollutant sink of the  $i$ th species is taken to be  $s - i(x, t) = a_0 n_i(x, t)$  for  $0 \leq x \leq x_0$ , and the pollutant source is  $s^+_i(x, t) = s^-_i(x - x_1, t)$ . The source-sink pair simulates rapid transport upward by  $x_1$ . We choose  $a_0 = 2.5 \times 10^{-5} \text{ s}^{-1}$  so that during the thunderstorm half of the density of the original plume is transported upward.

Fig. 3 shows the test results for this model at a time  $t = 4 \times 10^4$  s, when  $\approx 25\%$  of pollutants are transferred from the lower to the upper plume. The solid lines represent a simulation of the full chemical kinetics and the solid dots denote the spatially reduced model, with the relative tolerance parameter  $\delta = 10^{-2}$ . The gray

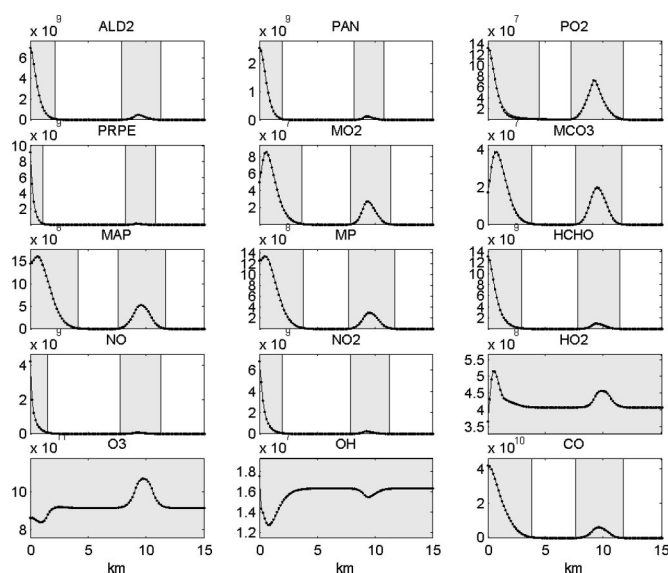
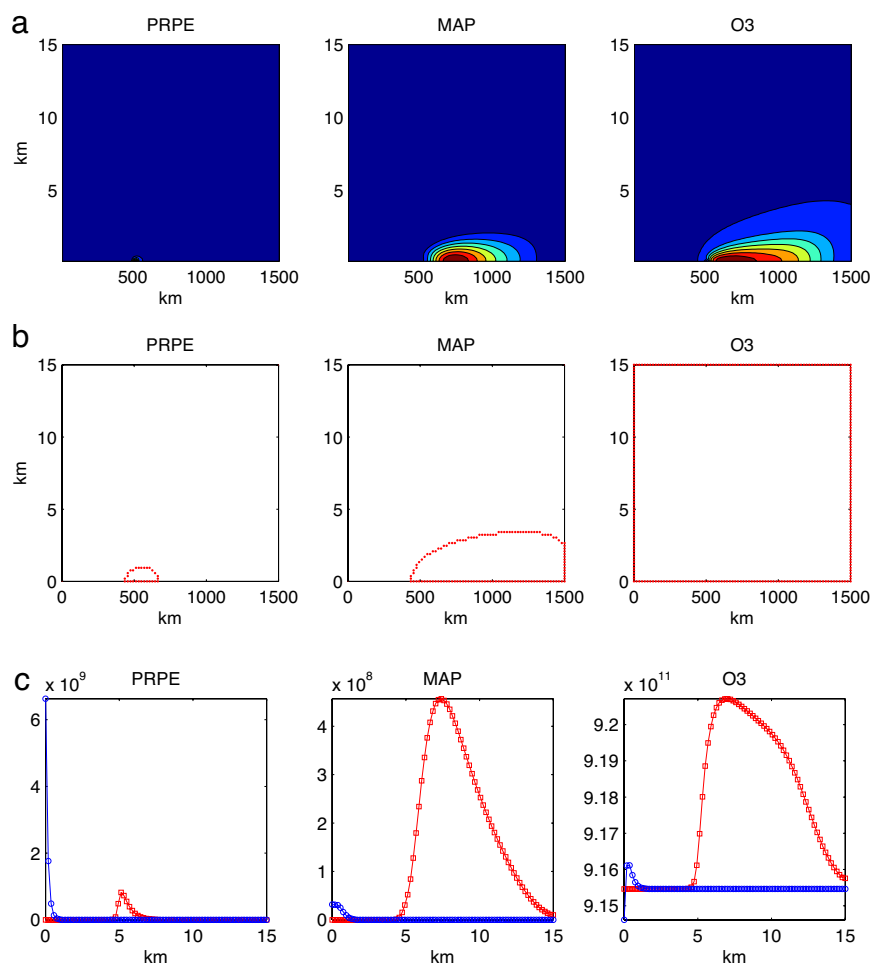


Fig. 3. Species densities versus distance at  $t = 4 \times 10^4$ . The computation is done in 15 km physical domain resolved by 100 points. Solid line, full chemical kinetics; dots, spatially reduced chemical kinetics.



**Fig. 4.** Comparison of spatially reduced and full calculations in two dimensions. (a) Concentrations of three selected species  $C_3H_6$ , MAP and  $O_3$  vs.  $(x, y)$ . (b) Moving boundary  $\Gamma$ , which separates the computational domain into two subdomains  $D_f$  and  $D_r$ , where full and reduced models are used accordingly. (c) Comparison of these species distribution obtained by the spatial reduction algorithm (spheres and cubes) and conventional method (solid line) depicted in blue and red colors at  $x = 750$  km and at  $y = 0.5625$  km correspondingly, at time  $t = 4 \times 10^4$  for the atmosphere with linearly distributed horizontal shear layer type flow. Vertical and horizontal spatial scales are  $y = 0 - 15$  km and  $x = 0 - 1,500$  km (not shown in the figure) correspondingly. The computation is carried out in  $1,500 \text{ km} \times 15 \text{ km}$  physical domain uniformly resolved by  $80 \times 80$  points.

regions in Fig. 3 denote the automatically generated domain partitioning. The test results demonstrate that the algorithm detects the changes in the source distribution, properly rearranges the domain splitting and maintains accuracy throughout the simulation.

It is worth noting that in the simulation all of the chemical concentrations are nonzero at every point in space, even in the slow regions where their governing reaction-diffusion-advection equations are not calculated explicitly. This is a subtle but important point for maintaining accuracy. For example, it is well known that there are chemicals such as nitrogen oxides (e.g., NO and  $NO_2$ ) in the reaction scheme that cannot be excluded from the chemical mechanism, even if their concentrations are very low.

What is the error in the spatial reduction? If we define the error as the norm of the discrepancy between the spatially reduced model  $n_i^0(x)$  and the full model  $n_i(x)$

$$e_i \equiv \|n_i(x) - n_i^0(x)\|_\infty / \|n_i^0(x)\|_\infty, \quad [12]$$

the largest error is obtained for  $PO_2$  density distribution  $e_{PO_2} = 3.1 \times 10^{-2}$ . Numerical experiments demonstrate that this error increases by a factor of five if the extrapolation algorithm in the slow region is not used, and the fast reactants are required to vanish in

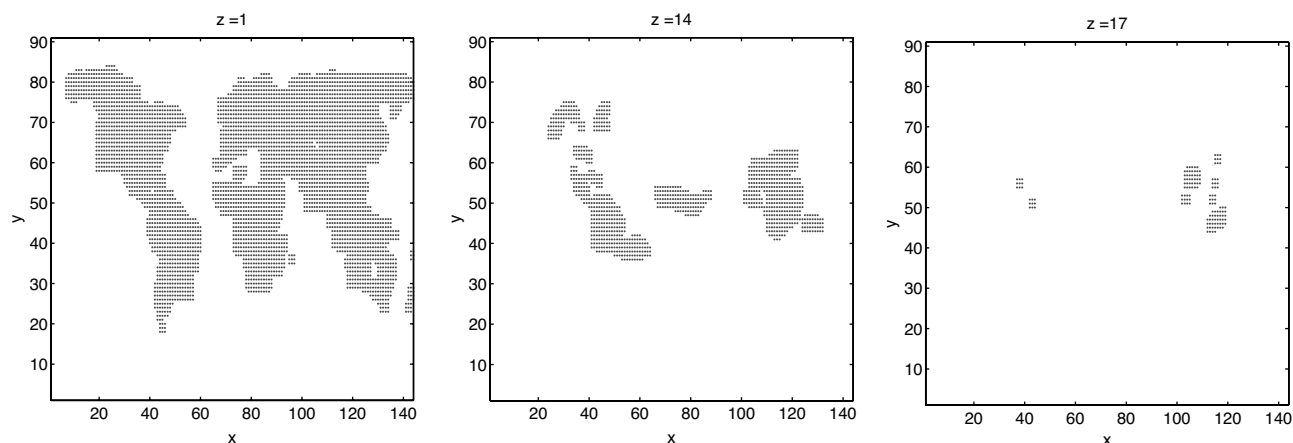
slow regions. Additionally, numerical experiments demonstrate that the error depends linearly on the thresholds  $\delta$ .

**Two-Dimensional.** Now we turn to a two dimensional simulation, where the extrapolation between fast and slow regions is not so simple. Our test case assumes that the horizontal and vertical components of the velocity field  $(u(x, y, t), v(x, y, t))$  form a shear layer,  $u(x, y) = u_0 + (u_1 - u_0)y/y_{\max}$  and  $v(x, y) = 0$ , where  $x$  and  $y$  are the horizontal and vertical coordinates and  $u_0 = 10$  m/s,  $u_1 = 30$  m/s. The simulation is carried out in a domain of dimensions  $1,500 \text{ km} \times 15 \text{ km}$ , with the computational domain uniformly resolved by  $80 \times 80$  points.

We assume the pollution source is two grid spacings wide, occurring between  $485 \text{ km} \leq x \leq 522 \text{ km}$ . Like the one-dimensional simulation above, the nonzero pollutant fluxes are  $J_{C_3H_6} = 10^{11}$  molecules per s  $cm^2$  and  $J_{NO} = 10^{11}$  molecules per s  $cm^2$ .

The molar concentrations of three selected species, PRPE, MAP, and  $O_3$ , with very different characteristic reaction times are shown in Fig. 4a at time  $t = 4 \times 10^4$ . The domains separating the fast and slow regions for these three species are shown in Fig. 4b.

Fig. 4c shows a quantitative comparison of the conventional technique (solid lines) and the spatial reduction algorithm (dots). The blue line shows a one dimensional slice at  $x = 750$  km, whereas



**Fig. 5.** Domain partitioning at the attitudes  $h = 0.17, 12.087$ , and  $15.198$  km.

the red line depicts a slice at  $y = 0.5625$  km. When  $\delta = 10^{-2}$  the largest error is  $e_{\text{PO}_2} = 3.7 \times 10^{-2}$ .

## Conclusions

The spatial reduction algorithm described herein allows great computational savings in the accurate simulation of chemical dynamics in the atmosphere. The algorithm relies on the fact that the concentration of fast reacting species decays rapidly with distance from the pollutant source. Therefore, for each chemical species, we partition the computational domain into two regions, where the full and reduced models are used accordingly. We have demonstrated the versatility of the method by applying it to a reduced mechanism for tropospheric oxidant chemistry. The algorithm works robustly and automatically in one and two spatial dimensions, with time-dependent or independent pollution sources.

Our current goal is to implement the algorithm into actual atmospheric solvers, while keeping the details simple enough that it can be easily implemented by the research community as a whole. The most difficult part of the implementation is the extrapolation between fast and slow regions. Although computationally efficient fast marching algorithms have been constructed for two- and three-dimensional problems (24–27), the technical difficulties associated with complex geometries shape of the moving separating interface could provide prohibitive for widespread implementation. For this reason, we have instituted a major simplification of our algorithm, in which we require that the domain boundaries sepa-

rating the fast and the slow regions have simple geometries. In our discussion above, we required the shapes of the domain boundaries to correspond to constant relative or absolute thresholds; however, there is no fundamental reason to do so. Increasing the size of the fast domain slightly can result in enormous simplification of the domain shape with only a slight increase in computational cost. The simplest possible shapes are rectangular patches made to be coincident with the existing grids used in current CTMs. We have adopted the fast marching algorithm for this geometry, and are currently working on adapting and implementing the algorithm for rectangular shapes in GEOS-Chem.

As an illustration, a three-dimensional domain splitting for the chemical PRPE (based on the PRPE density distribution produced by GEOS-Chem) is shown in Fig. 5. The computational mesh covering the fast subdomain is depicted by sets of dots at different horizontal layers. In the code, the earth's troposphere is uniformly resolved by  $91 \times 144$  points in horizontal  $x - y$  planes and by 20 layers of increasing thickness in vertical  $z$  direction. Fig. 5 depicts grid cross-sections at  $z = 1, 14$ , and  $17$ , corresponding to the attitudes  $h = 0.17, 12.087$ , and  $15.198$  km. The multiconnected fast subdomain is well captured although the rectangular patches; for example, Fig. 5a shows that the fast region essentially coincides with the locations of the continents, where the pollutant concentration is the highest.

This research was supported by the National Science Foundation Grant DMS-0417810.

- Thompson AM (1992) *Science* 256:1157–1165.
- Atkinson R, Baulch DL, Cox RA, Crowley JN, Hampson RF, Hynes RG, Jenkin ME, Rossi MJ, Troe J (2003) *Atmos Chem Phys Discuss* 3:6179–6699.
- Brasseur GP, Orlando JJ, Tyndall G (1999) *Atmospheric Chemistry and Global Change* (Oxford Univ Press, New York).
- Newell RE, Thouret V, Cho J, Stoller P, Marengo A, Smith H (1999) *Nature* 398:316–319.
- Chatfield RB, Brost RA (1987) *J Geophys Res* 92:13261–13276.
- Prather MJ, Jacob DJ (1997) *Geophys Res Lett* 24:3189–3192.
- Fiore AM, Jacob DJ, Bey I, Yantosca RM, Field BD, Fusco AC, Wilkinson JG (2002) *J Geophys Res* 107:4275.
- Singh NB, Jacob DJ (2000) *Atmos Environ* 34:4399–4401.
- Holmes CD, Jacob DJ, Yang X (2006) *Geophys Res Lett* doi:10.1029/2006GL027176.
- Bey I, Jacob DJ, Yantosca RM, Logan JA, Field B, Fiore AM, Li Q, Liu H, Mickley LJ, Schultz M (2001) *J Geophys Res* 106:23073–23096.
- Horowitz LW, Liang J, Gardner GM, Jacob DJ (1998) *J Geophys Res* 103:13451–13476.
- Logan JA, Prather MJ, Wofsy SC, McElroy MB (1981) *J Geophys Res* 86:7210–7254.
- Young TR, Boris JP (1977) *J Phys Chem* 81:2424–2427.
- Gong WM, Cho HR (1993) *Atmos Environ A* 27:2147–2160.
- Segel LA, Slemrod M (1989) *SIAM Rev* 31:446–477.
- Sandu A, Verwer J, Loon MV, Carmichael GR, Potra FA, Dabdub D, Seinfeld JH (1997) *Atmos Environ* 31:3149–3164.
- Maas U, Pope SB (1992) *Combust Flame* 88:239–264.
- Fraser S (1988) *J Chem Phys* 88:4732–4738.
- Singh S, Rastigejev Y, Paolucci S, Powers JM (2001) *Combust Theory Model* 5:163–184.
- Atkinson R (2000) *Atmos Environ* 34:2063–2101.
- Osher S, Sethian J (1988) *J Comp Phys* 79:12–49.
- Sethian JA (1999) *SIAM Rev* 41:199–235.
- Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) *Numerical Recipes* (Cambridge Univ Press, Cambridge, UK).
- Sethian JA, Adalsteinsson D (1997) *EEE Trans Semicond Manuf* 10:167–184.
- Malladi R, Sethian JA (1995) *Proc Natl Acad Sci USA* 92:7046–7050.
- Yoo S, Stewart DS (2005) *Combust Theory Model* 9:219–254.
- Rastigejev Y, Matalon M (2006) *Combust Theory Model* 10:459–481.
- Fiore A, Horowitz L, Dlugokencky E, West J (2006) *Geogr Res Lett* 33:L12809.